REMARKS

Claims 1-40 are pending in the present application. Applicants have amended Claims 19 and 37 herewith. Reconsideration of the claims is respectfully requested.

Applicants would initially like to thank the Examiner for taking the time to conduct a telephonic interview for this case on 10/29/2004. While no agreement was reached, Applicants' representative discussed the merits of the present invention, and in particular how the claimed communication management techniques were not taught by an internal garbage collection technique as described by the cited references.

I.   Objection to Specification

The Examiner objected to the Specification, stating the Abstract is too long. Applicants have amended the Abstract herewith to reduce the word count, as requested.

II.   35 U.S.C. § 102, Anticipation

The Examiner rejected Claims 1-5, 7-14, 16-17, 20-24, 26-32, 34-35 and 38-39 under 35 U.S.C. § 102(b) as being anticipated by Endicott et al. (U.S. 6,047,295). This rejection is respectfully traversed.

For a prior art reference to anticipate in terms of 35 U.S.C. 102, every element of the claimed invention must be identically shown in a single reference. *In re Bond*, 910 F.2d 831, 15 USPQ2d 1566 (Fed. Cir. 1990). Applicants will now show that every element of the claimed invention is not identically shown in a single reference, and thus these claims have been erroneously rejected under 35 USC 102(b).

With respect to Claim 1, such claim expressly recites (i) *responsive to conclusion of a communication process*, starting a timer; and (iii) *responsive to conclusion of a predetermined time period measured by the timer*, maintaining a weak reference to a connection object. Neither of features (i) or (ii) are taught by the cited Endicott reference, as will now be explained in detail.

Regarding missing claimed feature (i), such feature is directed to a particular step for managing, in a client, connections to a server and includes use of a timer which is started *responsive to the conclusion of a communication process*. The cited reference

does not teach such timer operation. In rejecting Claim 1, the Examiner cites Endicott Col. 7, line 66 – Col. 8, line 9). Applicants show that there, Endicott states:

> Any number of collection cycle indicators may be used to distinguish different collection cycles. As illustrated further herein, for example, a collection cycle indicator may be implemented as a cycle count that is maintained in a counter and incremented at the start of each collection cycle. The cycle count may have any range of values. For example, one suitable count is a single bit, which is capable of distinguishing between current and previous collection cycles. Other alternative cycle indicators may be used in the alternative, e.g., multi-bit counts, time stamps, and alphanumeric identifiers, among others.

As can be seen, this passage teaches that a counter is incremented at the start of each collection cycle. This collection cycle is with respect to an internal garbage collection scheme within a computer (Col. 1, line 48 – Col. 2, line 29; Col. 4, lines 39-63), and has nothing whatsoever to do with any type of communication process between a client and server. Thus, it is shown that Claim 1 has at least one missing claimed feature not taught by the cited reference.

As to missing claimed feature (ii), such feature is directed to changing from a normal reference to a weak reference for a connection object. In particular, *responsive to conclusion of a predetermined time period measured by the timer*, a weak reference is maintained to a connection object (which during the client connection to the server, had a normal reference to such connection object). The cited reference provides no details of any type of communication connection or its operation, only that such a connection may exist (Col. 5, lines 33-37). In addition, the timer cited by the Examiner in rejecting Claim 1 is not used in any fashion to change the type of reference that is maintained to a communication object, and in particular does not teach maintaining a weak reference to such a communication object responsive to conclusion of a predetermined time period measured by the timer. In fact, the cited reference makes no mention of any type of communication object whatsoever being used as a part of the (cursory-described) networked communication of Endicott. Even assuming arguendo that such communication object exists, there is no teaching or suggestion the a weak reference to such a (missing) communication object is maintained responsive to conclusion of a

predetermined time period measured by the timer. The timer as described by Endicott is merely used to count cycles as a part of an internal garbage collection technique, and is not used in any way as part of any communication between a client and server. Thus, Claim 1 is further shown to have been erroneously rejected as there is yet another missing claimed feature not taught by the cited reference.

Applicants initially traverse the rejection of Claims 2-5 and 7-12 for reasons given above with respect to Claim 1 (of which Claims 2-5 and 7-12 ultimately depend upon).

Further with respect to Claim 4 (and dependent Claim 6), Applicants urge that the cited reference does not teach the claimed features of determining whether a weak reference to the connection object exists; determining whether the connection object has been destroyed if the weak reference exists; and *reusing the connection if the connection object has not been destroyed.* The cited reference provides no details at all as to how a connection between a client and server is managed, and accordingly does not teach the specifics of *connection reuse if a connection object has not been destroyed.* In rejecting Claim 4, the Examiner cites Endicott Col. 10, line 48 – Col. 11, line 2. Applicants show that there, Endicott states:

> Another routine that may access weak references is dereference weak reference routine 100, which is illustrated in FIG. 5. Routine 100 uses a read barrier to ensure proper synchronization with the collector. Routine 100 is called in response to a request to obtain the pointer stored in a weak reference so that the information within the object referenced by the weak reference may be obtained through use of the pointer. Routine 100 begins in block 102 by determining whether the referenced object field for the weak reference is set to NULL, indicating that the weak reference is not pointing to another object. If so, block 104 is executed to return the NULL value and terminate the routine.
>
> If the referenced object is not NULL, control passes to block 106 to determine whether the cycle count for the weak reference is equal to the cycle count for the current collection cycle. If so, synchronization is assured by virtue of the object referenced by the weak reference being known strongly-reachable in the current collection cycle. Therefore, similar to set reference routine 88, the global weak reference locking mechanism may be bypassed by passing control to block 108 to return the pointer to the referenced object for the weak reference, whereby the routine is complete.

As can be seen, this passage merely describes a technique for returning a pointer stored in a weak reference so that information within the object referenced by the weak reference may be obtained by use of such pointer. This passage has nothing whatsoever to do with any type of connection reuse, or conditional connection reuse as determined by whether a connection object has been destroyed. Thus, Claim 4 (and dependent Claim 6) is further shown to have been erroneously rejected under 35 USC 102 as there are further missing claimed elements not taught by the cited reference.

Still further with respect to Claim 7, Applicants urge that the cited reference does not teach the claimed feature of "sending notification to the server that the connection object is unreferenced when a weak reference to the connection object is maintained". As can be seen, this claimed feature involves co-action with a server upon occurrence of a particular event. In rejecting Claim 7, the Examiner cites Endicott Col. 6, line 66 – Col. 7, line 15 as teaching this claimed feature. Applicants show that there, Endicott states:

> In the illustrated embodiment, weak references are managed using a concurrent mark sweep collector that operates in a collector thread concurrently with other program threads executing in the computer system. Performance of the computer system is improved by selectively inhibiting access by other program threads to only those weak references that reference objects not yet known to be strongly-reachable during the current collection cycle. Access to weak references that reference known strongly-reachable objects is permitted, however, since it is ensured that those strongly-reachable objects will not be later collected by the collector in the current collection cycle, and thus that the weak references will not be cleared by the collector during processing of the weak references. Consequently, there is no risk that a pointer retrieved from a weak reference during an access thereto will become invalid as a result of the collection process.

As can be seen, this passage describes management of weak object references, and in particular describes a technique of selectively inhibiting access by other program threads to only those weak references that reference objects not yet known to be strongly-reachable during a garbage collection cycle. This cited passage does not describe any type of server co-action technique, and in particular does not describe a step sending of a notification to a server that the connection object is unreferenced when a weak reference to the connection object is maintained. Therefore, Claim 7 is yet further shown to have

been erroneously rejected under 35 USC 102 as there is yet another claimed feature not taught by the cited reference.

Still further with respect to Claim 8, Applicants urge that the cited reference does not teach the claimed feature of "wherein the step of periodically destroying connection objects maintained by weak references comprises destroying the connection object in response to garbage collection by the server". As can be seen, this claim expressly recites that conditional step of periodic destroying of connection objects", with such destruction being in response to garbage collection by the server. In rejecting Claim 8, the Examiner cites Endicott Col. 3, lines 46-67. Applicants show that there, Endicott states:

> The invention addresses these and other problems associated with the prior art by providing a computer system, program product, and method of managing weak references with a concurrent mark sweep collector that, while processing weak references, inhibit access to only those weak references whose referenced objects have not yet been determined to be strongly-reachable (i.e., reachable other than through a weak reference) during a current collection cycle. Selective access inhibition in this manner is possible due to the realization that, once an object referenced by a particular weak reference is determined to be strongly-reachable during a current collection cycle, the object is ensured of not being collected during the current collection cycle. Moreover, the weak reference is also ensured of not being cleared when it is processed by the collector given its reference to a known strongly-reachable object. Additional synchronization for this weak reference is thus unnecessary for the remainder of the current collection cycle after the determination is made, so other program threads may be permitted to freely access this weak reference without waiting for the collector to complete the processing of all existing weak references.

As can be seen, this passage describes a selective access inhibition technique for internal garbage collection by a computer, so that other program threads may be permitted to access weak references without waiting for the garbage collector to complete processing of all existing weak references. Such passage does not describe (i) garbage collection for a client being done *by a server*, or (ii) any periodic destroying of *connection objects*. Thus, Claim 8 is yet further shown to have been erroneously rejected under 35 USC 102 as there is yet another claimed feature not taught by the cited reference.

With respect to Claim 13 (and dependent Claims 14, 16 and 17), Applicants traverse for similar reasons to the further reasons given above with respect to Claim 4.

With respect to Claim 20 (and dependent Claims 21-24 and 26-30), Applicants initially traverse for similar reasons to those given above regarding Claim 1.

Further with respect to Claim 23 (and dependent Claim 24), Applicants traverse for further reasons given above with respect to Claim 4.

Further with respect to Claim 26, Applicants traverse for further reasons given above with respect to Claim 7.

With respect to Claim 31 (and dependent Claims 32, 34 and 35), Applicants traverse for similar reasons to those given above with respect to Claim 13.

With respect to Claim 38, Applicants traverse for similar reasons to those given above with respect to Claim 1.

With respect to Claim 39, Applicants traverse for similar reasons to those given above with respect to Claim 13.

Therefore, the rejection of Claims 1-5, 7-14, 16-17, 20-24, 26-32, 34-35 and 38-39 under 35 U.S.C. § 102 has been overcome.

III.   35 U.S.C. § 103, Obviousness

A.   The Examiner rejected Claims 6, 15, 25 and 33 under 35 U.S.C. § 103 as being unpatentable over Endicott et al. (U.S. 6,047,295). This rejection is respectfully traversed.

Applicants traverse the rejection of Claims 6 and 25 for similar reasons to those given above regarding Claims 1 and 4, and show that the cited reference does not teach or suggest any method of managing connections, by a client, to a server, including the specific steps recited by such claims. The fact that a prior art device could be modified so as to produce the claimed device is not a basis for an obviousness rejection unless the prior art suggested the desirability of such a modification. *In re Gordon*, 733 F.2d 900, 221 USPQ 1125 (Fed. Cir. 1984). There is simply no desire suggested in the cited Endicott to modify the teachings therein in accordance with the invention recited in Claims 6 and 25. Thus, such claims are shown to have been erroneously rejected under 35 USC 103.

Applicants traverse the rejection of Claims 15 and 33 for similar reasons to those given above with respect to Claim 4, and urge that none of the cited references teach or suggest conditional reuse of a connection if a connection object has been destroyed. Thus, such claims are shown to have been erroneously rejected under 35 USC 103.

Therefore, the rejection of Claims 6, 15, 25 and 33 under 35 U.S.C. § 103 has been overcome.

B.   ·   The Examiner rejected Claims 18-19, 36-37 and 40 under 35 U.S.C. § 103 as being unpatentable over Endicott et al. (U.S. 6,047,295) in view of Nilsen et al. (U.S. 5,692,185). This rejection is respectfully traversed.

With respect to Claim 18 (and dependent Claim 19), none of the cited references teach or suggest the claimed step of "responsive to conclusion of a communication process using the connection, starting a timer", or "responsive to conclusion of a predetermined time period measured by the timer, removing the reference to the connection object from the hash map". In rejecting Claim 18, the Examiner states Endicott teaches starting a timer responsive to conclusion of a communication process using the connection at col. 7, line 67 – column 8, line 9. Applicants show that this passage merely teaches a counter that is incremented at the start of each collection cycle. This is different from what is claimed for at least two reasons. First, the claim recites that the counter is started responsive to *conclusion* of a *communication process*. Per the Endicott teachings, the counter is incremented at the start (not conclusion, as claimed) of each collection cycle (not communication process, as claimed). Thus, contrary to the Examiner's assertion, the cited Endicott reference does not teach the claimed step of "responsive to conclusion of a communication process using the connection, starting a timer".

Applicants further show that none of the cited references teach or suggest the claimed step of removing the reference to the connection object from the hash map responsive to conclusion of a predetermined time period measured by the timer. In rejecting Claim 18, the Examiner states on page 5 paragraph 6.a. of the Office Action dated 8/16/2004:

"Endicott *teaches* ... removing the reference to the connection object from the hash map responsive to conclusion of a predetermined time period measured by the timer (line 66 of column 2 through line 9 of column 3; and lines 56-59 of column 2)." (emphasis added by Applicants)

In the very next paragraph (1ˢᵗ sentence), the Examiner states:

"Endicott *fails to teach* ... removing the reference to the connection object from the hash map responsive to conclusion of a predetermined time period measured by the timer. " (emphasis added by Applicants)

Applicants agree with the later conclusion drawn by the Examiner (that Endicott fails to teach such reference removal step), and rebut the initial conclusion drawn by the Examiner (that Endicott teaches such reference removal step) by showing that at column 2, line 66 – column 3, line 9, Endicott discusses an ability to collect a shared data object through garbage collection techniques by using a weak, as opposed to strong, reference to such shared data object. There is no discussion of any hash map or use of a timer as part of such discussion.

Nor do the teachings of the cited Nilsen reference overcome such deficiency in teachings by Endicott. The Examiner states that "Miller describes, for example, a hashing function built into MultiScheme that associates a unique integer with each object (see reference given above). The hashing libraries retain a weak pointer to each object that has requested a hash number so that subsequent requests for the hash identity of the same object map to the same integer number," (lines 15-22 of column 44)". The Examiner then states "official notice" is taken of using two hash maps instead of one, and that it would have been obvious to combine the two hash maps having similar fields into one table with a single additional column for storing the state (hash or weak-hash) as an integer, in that the use of one hash map with an additional column is advantageous over using two hash maps in that it saves memory. Applicants have reproduced Claim 18 in its entirety below, for ease of discussion purposes:

Page 17 of 20
Tapperson – 09/844,340

A method, in a client, for caching connections to a server, comprising:

adding a reference to a connection object for a connection to a weak hash map and a hash map;

responsive to conclusion of a communication process using the connection, starting a timer; and

responsive to conclusion of a predetermined time period measured by the timer, removing the reference to the connection object from the hash map.

Applicants urge that none of the cited references teach or suggest the claimed step of "adding a reference to a connection object for a connection to a weak hash map and a hash map". As described above, the Examiner's position is that official notice is taken of using two hash maps instead of one. Applicants urge that Claim 18 is not merely directed to using two hash maps instead of one – but rather using two different types of hash maps, a 'hash map' and a 'weak hash map'. The Examiner has not alleged, nor do any of the cited references teach or suggest, the use of two *different types* of hash maps, or that a reference for a connection to a server, in particular a reference to a connection object, is added to two different types of hash maps. Thus, the Examiner has failed to establish a prima facie showing of obviousness with respect to Claim 18 (and dependent Claim 19), and such claim is thus further shown to have been erroneously rejected.

Applicants further show error in the rejection of Claim 18, in that none of the cited references teach or suggest the claimed step of *"responsive to conclusion of a predetermined time period measured by the timer,* removing the reference to the connection object from the hash map" (emphasis added by Applicants). Nor has the Examiner alleged any such teaching or suggestion. Thus, a prima facie case of obviousness has not been established with respect to Claim 18. Therefore, Claim 18 (and dependent Claim 19) is further shown to have been erroneously rejected.

Further with respect to Claim 19, none of the cited references teach or suggest the claimed steps of "determining whether the connection object has been destroyed"; and

"removing the reference to the connection object from the weak hash map if the
connection object has been destroyed". In rejecting Claim 19, the Examiner states
that such determination step is taught by Endicott at col. 7, lines 47-52. Applicants urge
that such passage makes no mention of any type of *connection* object, and thus does not
teach or suggest any type of determination of whether such (missing) connection object
has been destroyed. Applicants have amended Claim 19 to further emphasize the aspects
of such communication object. It is thus shown that Claim 19 is not obvious in view of
the cited references.

　　　　With respect to Claims 36-37 and 40, Applicants initially traverse for similar
reasons to those given above with respect to Claim 18.

　　　　Further with respect to Claim 37, Applicants traverse for further reasons given
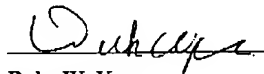above with respect to Claim 19.

　　　　Therefore, the rejection of Claims 18-19, 36-37 and 40 under 35 U.S.C. § 103 has
been overcome.

IV.   Conclusion

It is respectfully urged that the subject application is patentable over [the cited references and is now in condition for allowance. The Examiner is invited to call the undersigned at the below-listed telephone number if in the opinion of the Examiner such a telephone conference would expedite or aid the prosecution and examination of this application.

DATE: 11/3/04

Respectfully submitted,

Duke W. Yee
Reg. No. 34,285
Wayne P. Bailey
Reg. No. 34,289
Yee & Associates, P.C.
P.O. Box 802333
Dallas, TX 75380
(972) 385-8777
Attorneys for Applicant